UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/606,462	06/26/2003	Scott L. Michaelis	200205355-1	3496

22879 7590 12/08/2005

HEWLETT PACKARD COMPANY
P O BOX 272400, 3404 E. HARMONY ROAD
INTELLECTUAL PROPERTY ADMINISTRATION
FORT COLLINS, CO 80527-2400

EXAMINER	
SUGENT, JAMES F	
ART UNIT	**PAPER NUMBER**
2116	

DATE MAILED: 12/08/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/606,462	MICHAELIS ET AL.	
	Examiner	Art Unit	
	James Sugent	2116	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *26 June 2003*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☐ Claim(s) _____ is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-23* is/are rejected.

7)☒ Claim(s) *14 and 22* is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *23 June 2003* is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

Drawings

The drawings are objected to because:

- In Figure 4, reset code 410 in block 405 and slave CPU 410 in slave cell blocks 402b and 402c refer to different elements. As per Detailed Description reference in the Specification, please change slave CPU 410 in slave cell blocks 402b and 402c to slave CPU 415.
- In Figure 4, reset 415 in slave cell blocks 402b and 402c do not coincide with those element numbers referred to within the Detailed Description of the Specification. Please change reset 415 within slave cell blocks 402b and 402c to reset 416.

Corrected drawing sheets in compliance with 37 CFR 1.121(d) are required in reply to the Office action to avoid abandonment of the application. Any amended replacement drawing sheet should include all of the figures appearing on the immediate prior version of the sheet, even if only one figure is being amended. The figure or figure number of an amended drawing should not be labeled as "amended." If a drawing figure is to be canceled, the appropriate figure must be removed from the replacement sheet, and where necessary, the remaining figures must be renumbered and appropriate changes made to the brief description of the several views of the drawings for consistency. Additional replacement sheets may be necessary to show the renumbering of the remaining figures. Each drawing sheet submitted after the filing date of an application must be labeled in the top margin as either "Replacement Sheet" or "New Sheet"

pursuant to 37 CFR 1.121(d). If the changes are not accepted by the examiner, the applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

5

Specification

The disclosure is objected to because of the following informalities:

- Element numbers do not coincide with those previously used within the Specification or Drawings. Change "...slave cells 410b, 402c, which..." to
10 "...slave cells 402b, 402c, which..." (paragraph 23, line 5).

Appropriate correction is required.

Claim Objections

15 Claims 14 and 22 objected to because of the following informalities:

- Both claim 14 and 22 contain the use of the trademark/trade name ITANIUM or "Itanium Processor Family processors" as a limitation to identify or describe a particular material or product. The claim scope is uncertain since the trademark or trade name cannot be used properly to identify any particular material or product.
20 A trademark or trade name is used to identify a source of goods, and not the goods themselves. Thus, a trademark or trade name does not identify or describe

the goods associated with the trademark or trade name. Therefore, the claim is rendered indefinite.

Appropriate correction is required.

5

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

15

Claims 1-10, 12 and 13 is rejected under 35 U.S.C. 103(a) as being unpatentable over Haupt et al. (U.S. Patent No. 5,717,942) in view of Arndt (U.S. Patent No. 6,658,591 B1), Kubala (U.S. Patent No. 5,564,040) and Okuyama (U.S. Patent Publication No. 2002/0116469 A1).

20 As per **claim 1**, Haupt et al discloses a method for resetting a partition of a multiple partition system (100), wherein the partition (83 or 85) comprises a plurality of processors (partition 83 comprising processors 14, 16, 32 and 34; column 6, lines 5-10), the method comprising:

- executing reset code by an interfacing hardware (Haupt discloses an XBAR
25 interface block 86 that comprises means for resetting a partition that can be hardware or software triggered; column 6, line 63 thru column 7, line 5); and

- resetting one of the processors by writing a reset code to its associated reset register (column 9, line 66 thru column 10, line 16 and column 11, line 65 thru column 12, line 21).

Haupt does not explicitly disclose executing, by one processor of the plurality of processors, reset code from firmware; sending an interrupt to the other processors of the plurality of processors; and resetting the one processor by writing to its associated reset register.

Arndt teaches a method, system and apparatus for isolating fatal data fetch errors in a single partition system wherein the multiple processors (432, 434, 436 and 438) of the partition (430) are reset by code from firmware (hypervisor 410) by one of the of the processors (service processor) in the partition (column 11, lines 37-51). Arndt also teaches the sending of a system reset interrupt to the other processors in the partition (column 12, lines 34-37).

It would have been obvious to one of ordinary skill of the art, having the teachings of Haupt et al and Arndt before him at the time the invention was made, to modify reset method disclosed by Haupt et al to use the partition reset methods as taught by Arndt wherein one of the processors in the multiprocessor partition executes the reset code that is provided by firmware and a system reset interrupt is sent to the other processors in the partition.

One of ordinary skill in the art would be motivated to make use of the partition reset methods in view of the teachings of Arndt, as doing so would isolate fatal data fetch errors to a single partition and provide a plurality of data structure areas (Arndt: column 1, line 66 thru column 2, line 18).

Haupt et al and Arndt fail to teach building a list of reset register addresses associated with the plurality of processors.

Kubala teaches special interfaces defined for a server and a partition manager in a multiple partition system wherein when failure is detected in a partition, a partition manager (106) creates (packages up) a logout request that includes a list of valid processors and their prefix register values . Once the server completes the logout, the partition manger (106) resets the partition (column 7, lines 39-50).

It would have been obvious to one of ordinary skill of the art, having the teachings of Haupt et al, Arndt and Kubala before him at the time the invention was made, to modify the partition reset method disclosed by Haupt et al and Arndt to create a list of valid processors and prefix register values as taught by Kubala such that the list of valid processors and prefix register values comprises a list of reset register addresses associated with the plurality of processors before resetting the partition.

One of ordinary skill in the art would be motivated to make use of reset register address list creation in view of the teachings of Kubala, as doing so would incorporate protective measures to prevent tampering with hardware and software facilities (column 1, lines 49-52).

Haupt et al, Arndt and Kubala fail to teach resetting the other processors by writing a reset code to their associated reset registers.

Okuyama teaches a multiprocessor system (system 100 comprising processors 10-1, 10-2...10-n) with a shared-memory method wherein all of the processors in the system are reset by writing a code in the reset register (paragraphs 140-142).

It would have been obvious to one of ordinary skill of the art, having the teachings of Haupt et al, Arndt, Kubala and Okuyama before him at the time the invention was made, to

modify the multiprocessor reset method disclosed by Haupt et al, Arndt and Kubala to use the reset register coding as taught by Okuyama wherein code will be set to all of the processors within a multiprocessor system to reset themselves.

One of ordinary skill in the art would be motivated to make use of reset register coding in view of the teachings of Okuyama, as doing so would include a recovery process for recovering the failure in updating data in shared memory (paragraph 15).

As per **claim 2**, Arndt teaches a method further comprising: storing the firmware on a read only memory (Arndt: 6, 25-32).

As per **claim 3**, Arndt teaches a method further comprising: storing the list of reset register addresses in random access memory (Arndt teaches an error log to report fatal errors to be saved on non-volatile random access memory NV-RAM; column 11, 38-51).

As per **claim 4**, Arndt teaches a method wherein the last reset register address on the list is a reset register address for the cell (cluster) that is executing the reset code (Arndt teaches resetting the other processors in the partition after creation of an error log then resetting [rebooting] the calling processor using reset registers; column 9, lines 1-5 and column 11, lines 38-51).

As per **claim 5**, Arndt teaches a method further comprising: requesting the execution of the reset code by a processor of the plurality of processors (Arndt: column 12, lines 29-37).

As per **claim 6**, Arndt teaches a method further comprising: requesting the execution of the reset code by an operating system of the multiple partition system (Arndt: column 9, lines 1-10).

As per **claim 7**, Arndt teaches a method further comprising: requesting the execution of the reset code by a firmware shell (Arndt: hypervisor 410) of the multiple partition system (Arndt: column 9, lines 1-5).

As per **claim 8**, Kubala teaches a method wherein building the list comprises: writing the address of the one processor last in the list (Kubala teaches a logical partition manager creating a list of register addresses wherein logout is performed then reset of the partition followed by load of server code to restart the service processor last; column 7, lines 39-51).

As per **claim 9**, Okuyama teaches a method further comprising: flushing a cache associated with the one processor, after sending the interrupt (Okuyama: paragraph 140).

As per **claim 10**, Arndt teaches a method further comprising: moving execution from main memory to read only memory (Arndt teaches the effected processor moving execution to from main memory to machine check firmware resident on the processor; column 8, lines 38-48).

As per **claim 12**, Haupt teaches a method wherein the partition comprises a plurality of cells (Haupt: clusters [all hardware associated with controllers 12 and 26] associated with partitions 83 and 85), and each cell comprises at least one processor (Haupt: processors 14 and 16 for controller 12), the method further comprises: inventorying the plurality of cells for resetting (Haupt teaches the reset process of a partition which is handled by an XBAR interface block 86 that determines [inventories hardware/partitions available] what is to be processed; column 9, line 54 thru column 10, line 16).

As per **claim 13**, Kubala teaches a method wherein resetting the one processor occurs after resetting the other processors (Kubala teaches a logical partition manager creating a list of register addresses wherein logout is performed then reset of the partition followed by load of server code to restart the service processor last; column 7, lines 39-51).

Claim 11 is rejected under 35 U.S.C. 103(a) as being unpatentable over Haupt et al. (U.S. Patent No. 5,717,942), Arndt (U.S. Patent No. 6,658,591 B1), Kubala (U.S. Patent No. 5,564,040) and Okuyama (U.S. Patent Publication No. 2002/0116469 A1) as applied to claim 1 above, and further in view of Patterson et al. (Patterson, David A, et al., "Computer Organization and Design", Morgan Kaufman Publishers, Inc., 1998, pp. 579-583).

As per **claim 11**, Haupt et al, Arndt, Kubala and Okuyama fail to teach a method further comprising: switching from virtual memory mode into physical memory mode.

Patterson et al teaches of computer systems using virtual memory wherein the active programs must be present in main memory as is inherent for all systems using virtual memory

such that active programs are switched from virtual memory (disk memory) into physical memory (main memory) (pg. 579, line 30 thru pg. 580, line 7).

It would have been obvious to one of ordinary skill of the art, having the teachings of Haupt et al, Arndt, Kubala, Okuyama and Patterson et al before him at the time the invention was made, to modify the memory system disclosed by Haupt et al, Arndt, Kubala and Okuyama to use the virtual memory system as taught by Patterson et al wherein active programs will be moved from virtual memory to main memory.

One of ordinary skill in the art would be motivated to make use of a virtual memory system as in the view of the teachings of Patterson et al, as doing so would allow efficient and safe sharing of memory among multiple programs and to remove the programming burdens of a small, limited amount of main memory (pg, 579, lines 32-34).

Claim 14 is rejected under 35 U.S.C. 103(a) as being unpatentable over Haupt et al. (U.S. Patent No. 5,717,942), Arndt (U.S. Patent No. 6,658,591 B1), Kubala (U.S. Patent No. 5,564,040) and Okuyama (U.S. Patent Publication No. 2002/0116469 A1) as applied to claim 1 above, and further in view of Schelling (U.S. Patent Publication No. 2003/0229775 A1).

As per **claim 14**, Haupt et al, Arndt, Kubala and Okuyama fail to teach a method wherein the firmware is system abstraction layer firmware.

Schelling teaches a multiprocessor system (100) wherein the firmware associated with the system comprises a system abstraction layer (paragraphs 16 and 17).

It would have been obvious to one of ordinary skill of the art, having the teachings of Haupt et al, Arndt, Kubala, Okuyama and Schelling before him at the time the invention was

made, to modify the firmware memory system disclosed by Haupt et al, Arndt, Kubala and Okuyama to include the system abstraction layer as taught by Schelling.

One of ordinary skill in the art would be motivated to make use of a virtual memory system as in the view of the teachings of Schelling, as doing so would maintain a healthy status processors when system errors are detected and reset is necessary (paragraphs 2 and 11).

Claims 15-21 are rejected under 35 U.S.C. 103(a) as being unpatentable over Haupt et al. (U.S. Patent No. 5,717,942) in view of Arndt (U.S. Patent No. 6,658,591 B1) and Kubala (U.S. Patent No. 5,564,040).

As per **claim 15**, Haupt et al discloses a partition of multiple partition computer system comprising: a plurality of processors (partition 83 comprising processors 14, 16, 32 and 34; column 6, lines 5-10).

Haupt does not explicitly teach firmware comprising reset code that resets a portion of the partition, wherein one processor of the plurality of processors executes the reset code.

Arndt teaches a method, system and apparatus for isolating fatal data fetch errors in a single partition system wherein the multiple processors (432, 434, 436 and 438) of the partition (430) are reset by code from firmware (hypervisor 410) by one of the of the processors (service processor) in the partition (column 11, lines 37-51).

It would have been obvious to one of ordinary skill of the art, having the teachings of Haupt et al and Arndt before him at the time the invention was made, to modify reset method disclosed by Haupt et al to use the partition reset methods as taught by Arndt wherein one of the processors in the multiprocessor partition executes the reset code that is provided by firmware.

One of ordinary skill in the art would be motivated to make use of the partition reset methods in view of the teachings of Arndt, as doing so would isolate fatal data fetch errors to a single partition and provide a plurality of data structure areas (Arndt: column 1, line 66 thru column 2, line 18).

Haupt et al and Arndt fail to teach random access memory that is not affected by the reset code, that stores a list of addresses associated with the portion.

Kubala teaches special interfaces defined for a server and a partition manager in a multiple partition system wherein when failure is detected in a partition, a partition manager (106) creates (packages up) a logout request that includes a list of valid processors and their prefix register values. Once the server completes the logout, the partition manger (106) resets the partition (column 7, lines 39-50).

It would have been obvious to one of ordinary skill of the art, having the teachings of Haupt et al, Arndt and Kubala before him at the time the invention was made, to modify the partition reset method disclosed by Haupt et al and Arndt to create a list of valid processors and prefix register values as taught by Kubala such that the list of valid processors and prefix register values comprises a list of reset register addresses associated with the plurality of processors and is stored in the NV-RAM (non-volatile memory) error log taught by Arndt (column 8, lines 38-67 and column 11, lines 46-51).

One of ordinary skill in the art would be motivated to make use of reset register address list creation in view of the teachings of Kubala, as doing so would incorporate protective measures to prevent tampering with hardware and software facilities (column 1, lines 49-52).

As per **claim 16**, Arndt teaches a partition of multiple partition computer system further comprising: read only memory that stores the firmware (Arndt: column 6, lines 25-32).

As per **claim 17**, Haupt teaches a partition of multiple partition computer system further comprising:

- a plurality of cells (Haupt: clusters associate with controllers 12, 26, 28 and 30);
- wherein each cell (Haupt: cluster associate with a controller) comprises at least one processor (Haupt: processors 14 and 16 associated with controller 12; processors 32 and 34 associated with controller 26; etc.) of the plurality of processors, and each cell comprises a reset register (Haupt: reset registers RST 120, 150, 170 and 188) having an address that is on the list (Haupt: column 8, lines 36-50 and column 9, lines 25-53).

As per **claim 18**, Haupt teaches a partition of multiple partition computer system wherein each cell (Haupt: cluster associate with a controller) is reset by writing a reset code to its associated reset register (reset registers 120, 150, 170 and 188; column 9, line 66 thru column 10, line 16).

As per **claim 19**, Haupt teaches a partition of multiple partition computer system wherein the address on the list is a reset register address (Haupt: 120) for the cell (Haupt: cluster associate with a controller 12) that comprises the one processor (Haupt teaches the means for passing the

reset code from the controller 122 from the affected processor; column 9, line 66 thru column 10, line 46 and column 11, line 66 thru column 12, line 21).

As per **claim 20**, Haupt teaches a partition of multiple partition computer system wherein each cell (Haupt: cluster associate with a controller 12) further comprises resources other than the at least one processor (Haupt: XBAR interface block 86), and a portion of the resources is reset when the cell is reset (Haupt: column 12, lines 9-21).

As per **claim 21**, Kubala teaches a partition of multiple partition computer system wherein the one processor is reset after the other processors of the plurality of processors are reset (Kubala teaches a logical partition manager creating a list of register addresses wherein logout is performed then reset of the partition followed by load of server code to restart the service processor last; column 7, lines 39-51).

Claim 22 is rejected under 35 U.S.C. 103(a) as being unpatentable over Haupt et al. (U.S. Patent No. 5,717,942), Arndt (U.S. Patent No. 6,658,591 B1) and Kubala (U.S. Patent No. 5,564,040) as applied to claim 15 above, and further in view of Schelling (U.S. Patent Publication No. 2003/0229775 A1).

As per **claim 22**, Haupt et al, Arndt and Kubala fail to teach a partition of multiple partition computer system wherein the firmware is system abstraction layer firmware.

Schelling teaches a multiprocessor system (100) wherein the firmware associated with the system comprises a system abstraction layer (paragraphs 16 and 17).

It would have been obvious to one of ordinary skill of the art, having the teachings of Haupt et al, Arndt, Kubala and Schelling before him at the time the invention was made, to modify the firmware memory system disclosed by Haupt et al, Arndt and Kubala to include the system abstraction layer as taught by Schelling.

One of ordinary skill in the art would be motivated to make use of a virtual memory system as in the view of the teachings of Schelling, as doing so would maintain a healthy status processors when system errors are detected and reset is necessary (paragraphs 2 and 11).

Claim 23 is rejected under 35 U.S.C. 103(a) as being unpatentable over Haupt et al. (U.S. Patent No. 5,717,942) in view of Arndt (U.S. Patent No. 6,658,591 B1), Kubala (U.S. Patent No. 5,564,040) and Okuyama (U.S. Patent Publication No. 2002/0116469 A1).

As per **claim 23**, Haupt et al discloses a computer readable medium (XBAR interface logic block 86) having computer program logic recorded thereon (column 9, lines 15-24) for operating a partition of a multiple partition computer system (100), wherein the partition (83 or 85) comprises a plurality of processors (partition 83 comprising processors 14, 16, 32 and 34; column 6, lines 5-10).

Haupt does not explicitly disclose means for placing each processor of the plurality of processors into a known state.

Arndt teaches a method, system and apparatus for isolating fatal data fetch errors in a single partition system wherein the multiple processors (432, 434, 436 and 438) of the partition (430) are reset by code from firmware (hypervisor 410) by one of the of the processors (service processor) in the partition (column 11, lines 37-51). Arndt also teaches a means for (by the

effected processor) sending of a system reset interrupt to the other processors in the partition (they are placed in an interrupted state; column 12, lines 34-37).

It would have been obvious to one of ordinary skill of the art, having the teachings of Haupt et al and Arndt before him at the time the invention was made, to modify reset method disclosed by Haupt et al to use the partition reset methods as taught by Arndt wherein one of the processors in the multiprocessor partition executes a system reset interrupt is sent to the other processors in the partition.

One of ordinary skill in the art would be motivated to make use of the partition reset methods in view of the teachings of Arndt, as doing so would isolate fatal data fetch errors to a single partition and provide a plurality of data structure areas (Arndt: column 1, line 66 thru column 2, line 18).

Haupt et al and Arndt fail to teach building means for building a list of reset register addresses associated with the plurality of processors.

Kubala teaches special interfaces defined for a server and a partition manager in a multiple partition system wherein when failure is detected in a partition, Kubala provides means for (partition manager 106) creating (packaging up) a logout request that includes a list of valid processors and their prefix register values . Once the server completes the logout, the partition manger (106) resets the partition (column 7, lines 39-50).

It would have been obvious to one of ordinary skill of the art, having the teachings of Haupt et al, Arndt and Kubala before him at the time the invention was made, to modify the partition reset method disclosed by Haupt et al and Arndt to create a list of valid processors and

prefix register values as taught by Kubala such that the list of valid processors and prefix register values comprises a list of reset register addresses associated with the plurality of processors before resetting the partition.

One of ordinary skill in the art would be motivated to provide means for reset register address list creation in view of the teachings of Kubala, as doing so would incorporate protective measures to prevent tampering with hardware and software facilities (column 1, lines 49-52).

Haupt et al, Arndt and Kubala fail to teach means for resetting the plurality of processors by writing a reset code into their associated reset registers.

Okuyama teaches a multiprocessor system (system 100 comprising processors 10-1, 10-2...10-n) with a shared-memory method which provides means for (interruption handler program 36) resetting all of the processors in the system by writing a code in the reset register (paragraphs 140-142).

It would have been obvious to one of ordinary skill of the art, having the teachings of Haupt et al, Arndt, Kubala and Okuyama before him at the time the invention was made, to modify the multiprocessor reset method disclosed by Haupt et al, Arndt and Kubala to provide means for use of reset register coding as taught by Okuyama wherein code will be set to all of the processors within a multiprocessor system to reset themselves.

One of ordinary skill in the art would be motivated to make use of reset register coding in view of the teachings of Okuyama, as doing so would include a recovery process for recovering the failure in updating data in shared memory (paragraph 15).

Any inquiry concerning this communication or earlier communications from the examiner should be directed to James Sugent whose telephone number is (571) 272-5726. The examiner can normally be reached on 8AM - 4PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Lynne Browne can be reached on (571) 272-3670. The fax phone number for the organization where this application or proceeding is assigned is (571) 273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at (866) 217-9197 (toll-free).

James Sugent
Patent Examiner, Art Unit 2116
November 28, 2005